Working with getElementById in JavaScript

From: http://coursesweb.net/javascript/getelementbyid

The ID is an attribute that can be added to HTML tags. The value added at this attribute assigns an unique ID for that HTML tag. The "id" can be used to define CSS styles, but can also be used in JavaScript scripts to work with HTML elements and their content. Thus, the id can make a connection between JavaScript and HTML tags.
To reference in a JS script to a HTML tag through its ID, use the following syntax:

```
document.getElementById("id")
```

- **getElementById()** is a method of document object, it gets the tag element with the value "id" in its ID attribute.
- - You can use single quotes ('') for id in parentheses.

This syntax ">i>document.getElementById ("id")" returns a reference to the whole HTML element that has thad "id ", define an object containing its attributes and content. To refer to some parts of this object (HTML element), eg. its content or an "style" attribute, you must use specific properties of this object. Below is a list of the most used.

Properties that can be used with getElementById

- attributes[] - contains an Array (index starting from 0) all attributes of that HTML tag. It has two properties:
    - name - returns the attribute name
    - value - returns the attribute value
- childNodes - contains an array of all the elements directly underneath the current object.
    - For example, if you want to get the text content of a HTML tag, you can use the fallowing code:
      - ***document.getElementById("id").childNodes[0].nodeValue***
- className - gets or sets the value of "class" attribute
  Example: to get - var cls=object.className; to set - object.className='value';
- getAttribute("attribute") - gets the value of the attribute in parameter
- setAttribute("attribute", "value") - sets or changes the value of an attribute
- removeAttribute("attribute") - Removes the specified attribute
- href - gets or sets the value of "href" attribute (used for <a> tags)
- innerHTML - sets or retrieves the text HTML content
- src - gets or sets the value of "src" attribute
- style - defines values of the "style" attribute, used for CSS style elements. This property is followed by a CSS property.
- value - used for Form elements (from <form>), sets or gets the value of a form element (input text-box, button, ...)

To understand how to work with "document.getElementById ("id")" and the properties listed above, here are some examples with each of them.

1. attributes

In this example we use a JavaScript function that is executed when we click on a link, it will display an alert window with the name of the seccond attribute of the <a> tag.

```
<script type="text/javascript"><!--
function test(val) {
  var attr = val.attributes[1].name;
  alert(attr);
}
--></script>
<a href="#" id="id_link" title="test link" onclick="test(this)">A link</a>
```

Notice that it is not used "document.getElementById('an_id')", but it's used the "this" property. It refers to the current element. But you could also use "*document.getElementById('id_link').attributes[1].name*" instead of "val.attributes[1].name", the result will be the same.

2. childNodes

Working with requires knowledge of several properties belonging to it, but here it is presentef a simple example to understand how to use it.
In this example we have a DIV that has some text and a SPAN tag. To see the results of the script with "childNodes", we use a link to call a function that gets the text content of the SPAN tag.

```
<script type="text/javascript"><!--
function test2() {
  var val2 = 
document.getElementById("ex2").childNodes[1].childNodes[0].nodeValue;
  alert(val2);
}
--></script>
<div id="ex2">Web courses - <span>http://coursesweb.net</span> - lessons, 
tutorials</div>
<a href="#" onclick="test2()" title="childNodes example">Test 2</a>
```

- document.getElementById("ex2").childNodes[1].childNodes[0] gets the first element (childNodes[0]) of the second element (childNodes[1]) of the HTML tag with id="ex2".

3. getAttribute() and setAttribute()

In this example we use two links that call a JavaScript function, with different parameters.
The function gets an alerts a message with the "title" attribute value of the tag with id="ex3", but when the parameter has the value of 1, this function sets another value for that "tittle" attribute.

```
<script type="text/javascript"><!--
function test3(nr) {
  if (nr == 1) {
    document.getElementById("ex3").setAttribute('title', 'title changed');
  }
  var val3 = document.getElementById("ex3").getAttribute('title');
  alert(val3);
}
--></script>
<a href="#" id="ex3" title="getAttribute() example" onclick="test3(0)">Link 
1</a> - <a href="#" title="Link 2 - setAttribute()" onclick="test3(1)">Link 
2</a>
```

When you want to change the value of "class" attribute, use ClassName, because IE causes problems with "setAttribute" when it's used to classes.
Example:

```
  object.className = 'value';
```

4. href

Here is an example that modifies the URL added in the "href" attribute of a link.

```
<script type="text/javascript"><!--
function test4() {
  document.getElementById("ex4").href="http://coursesweb.net/javascript/";
}
--></script>
<a href="http://coursesweb.net/ajax/" id="ex4" title="href example"
target="_blank" onmouseover="test4()">Ajax</a>
```

- The link has set the " http://coursesweb.net/ajax/ " URL, but when the mouse is over the link, "onmouseover" calls the test4() that changes the value of "href" attribute. When you click on it will open the page: http://coursesweb.net/javascript/ .

• You can proceed in the same way FOR "src", which refers to the 'src' attribute (in <img> tags).

5. innerHTML

With **innerHTML** property you can get the content surrounded by an HTML tag, but it can also replace that content with another one, that can contains also HTML tags.
In this example we have a JS function, a DIV tag and two links. The first link calls the function to display an Alert with the DIV content. The second link transfers a parameter to the same function (its "title" tag value) that indicate it to change the DIV content with that parameter.

```
<script type="text/javascript"><!--
function test5(vl) {
  if (vl == 0) {
    var val5 = document.getElementById("ex5").innerHTML;
    alert(val5);
  }
  else {
    document.getElementById("ex5").innerHTML = vl;
  }
}
--></script>
<div id="ex5">Web courses - <span>coursesweb.net</span> - lessons,
tutorials</div>
<a href="#" title="innerHTML example" onclick="test5(0)">Show Alert</a> - <a
href="#" title="innerHTML - replacement"
onclick="test5(this.getAttribute('title'))">Change the DIV content</a>
```

- when the "document.getElementById("ex5").innerHTML" expresion is in the right side of the "=" character, it gets the tag content, but when it's in the left side of the "=" character, adds into the tag the value written after the '=' (in right side).

6. style

The "**style**" property can change the graphic style of the elements, using CSS properties. The syntax is:

- **document.getElementById("id").style.cssProperty = "value";**

- "cssProperty" and "value" are taken from CSS. Ex.: color (the property) and blue (the value).
The difference appears when are used composed CSS properties, like "font-weight", "margin-top" or "border-top-width", ... In JavaScript disappears the dash '-' character and the next words are written with uppercase first character, so for "font-weight" from CSS in JS is "fontWeight", and

"border-top-width" in JS is "borderTopWidth".
Example:

```
<script type="text/javascript"><!--
function test6(nr) {
  if (nr == 0) {
    document.getElementById('ex6').style.display = 'inline';
  }
  else if (nr == 1) {
    document.getElementById('ex6').style.color = '#0101ff';
    document.getElementById('ex6').style.fontWeight = 'bold';
  }
}
--></script>
<div>
  <a href="#" title="style ex" onmouseover="test6(0)">Link ex6</a>
  - <span id="ex6" style="display:none;" onmouseover="test6(1)">JS
getElementById()</span>
</div>
```

- When the test6() function is called with a parameter that has the value of 0, it changes the CSS "display" property of the <span> tag to "inline", making it visible.
- If the parameter is 1, the function sets / changes the value of "color" and "fontWeight" properties.

7. value

The **value** property is used with Form elements, it gets or sets values to fields of the form.
- To use "*getElementById('id')*" with form elements, they must have assigned an "id".
Here's a simple example that displays an Alert with the text written in a text box.

```
<script type="text/javascript"><!--
function test7() {
  var val7 = document.getElementById("ex7").value;
  alert(val7);
}
--></script>
<form action="" method="post">
 <input type="text" id="ex7" />
 <input type="button" value="Click" onclick="test7()" />
</form>
```